file.py

Overview:

This Python file provides a set of basic arithmetic functions. It is designed as a simple utility module to perform addition, subtraction, multiplication, and division. Each function is self-contained and takes two numerical inputs to produce a single numerical output. The primary purpose of this file is to offer fundamental mathematical operations that can be easily imported and used in other parts of a larger application.

FunctionDef add

This function performs the arithmetic operation of addition. It takes two numerical arguments, `a` and `b`, and computes their sum. The function is straightforward and directly returns the result of the addition. It can handle various numerical types, including integers and floating-point numbers.

Parameters: * `a`: The first number in the addition. It can be an integer or a float. * `b`: The second number in the addition. It can be an integer or a float.

Returns: The sum of `a` and `b`. The return type will be an integer if both inputs are integers, and a float if at least one of the inputs is a float.

Note: This function does not perform any type checking. If non-numerical types are passed as arguments, a `TypeError` will be raised.

Examples:

```python

# Example 1: Adding two integers

add(5, 3) 8

# Example 2: Adding an integer and a float

add(5.5, 3) 8.5 ```

## FunctionDef subtract

This function performs the arithmetic operation of subtraction. It takes two numerical arguments, `a` and `b`, and calculates the difference by subtracting `b` from `a`. The function returns this result. It is capable of handling both integer and floating-point numbers.

**Parameters**: * `a`: The minuend, or the number from which another number is to be subtracted. It can be an integer or a float. * `b`: The subtrahend, or the number that is to be subtracted. It can be an integer or a float.
```

Returns: The difference between a and b. The return type will be an integer if both inputs are integers, and a float if at least one of the inputs is a float.

Note: Similar to the add function, this function does not validate the input types. Providing non-numerical arguments will result in a TypeError.

Examples:

~~~python

# Example 1: Subtracting two integers

subtract(10, 4) 6

# Example 2: Subtracting a float from an integer

subtract(10, 4.5) 5.5 ```

## FunctionDef multiply

This function performs the arithmetic operation of multiplication. It accepts two numerical arguments, a and b, and computes their product. The resulting value is then returned. The function works with both integers and floating-point numbers.

**Parameters**: * a: The first number in the multiplication (multiplicand). It can be an integer or a float. * b: The second number in the multiplication (multiplier). It can be an integer or a float.

**Returns**: The product of a and b. The return type will be an integer if both inputs are integers, and a float if at least one of the inputs is a float.

**Note**: This function will raise a TypeError if arguments are not numerical types.

**Examples:**

```python

# Example 1: Multiplying two integers

multiply(7, 6) 42

# Example 2: Multiplying an integer by a float

multiply(7, 0.5) 3.5 ```
~~~

FunctionDef divide

This function performs the arithmetic operation of division. It takes two numerical arguments, `a` (the dividend) and `b` (the divisor), and returns the result of `a` divided by `b`. In Python 3, this function always performs float division, meaning the result will be a float even if the division is exact.

Parameters: * `a`: The dividend, or the number to be divided. It can be an integer or a float. * `b`: The divisor, or the number by which to divide. It can be an integer or a float.

Returns: The quotient of `a` divided by `b`. The return type is always a float.

Note: This function has a critical edge case. If the divisor `b` is zero, the function will raise a `ZeroDivisionError`. It is the responsibility of the caller to handle this potential exception.

Examples:

```python

## Example 1: Dividing two integers

divide(20, 5) 4.0

## Example 2: Division by zero

divide(10, 0)

## Raises ZeroDivisionError

```

Called_functions:

The functions within this file (`add`, `subtract`, `multiply`, `divide`) are self-contained and do not call any other functions defined in this script or from external libraries. They each perform a single, distinct arithmetic calculation using basic Python operators (`+`, `-`, `*`, `/`) on the provided input parameters.